Amendments to the Specification:

Please amend the specification as follows:

On page 1, line 2, please change the title of the invention to read -- A METHOD OF SEARCHING FOR AND RETRIEVING INFORMATION FROM STRUCTURE DOCUMENTS --

On page 1 paragraph beginning at line 10, amend as follows:

B1 It is known in the art to extract necessary elements from a plurality of documents and deal with them for generating an output document. A conventional editing technique for obtaining such an output document is disclosed in Japanese Laid-open ~~Paten~~ Patent Application No. 6-259421. According to this conventional technique, the elements of the structured input document are extracted using matches of character sequences with the elements in the document, a sequence connector, a hierarchy connector, etc.

On page 2, paragraph beginning on line 5, please amend as follows:

B2 In brief, ~~these objects are~~the object is achieved by a technique of editing a plurality of structured documents ~~is disclosed~~. A plurality of structured documents are inputted in a document edit system. Thereafter, a plurality of elements are extracted from each of the plurality of structured documents using an element edit statement which indicates element to be extracted. The extraction of the elements is implemented while the relationship of the elements extracted is maintained.

On page 3, paragraph beginning on line 24, please amend as follows:

B3 Tags are ~~is~~ delimited using the '<' and '>' characters. Tags are used to define an element which is an identified component of a document. The element usually consists of a start-tag, content and an end-tag. However, an element may involve one or more elements. The end-tag is delimited by '</'and'>'. Element matching operation using the tag is briefly described with reference to Fig. 1. As shown in Fig. 1, an element edit statement 10 includes a tag <note>. A reference numeral 12 denotes a structured document. The tag <note> in the

B3 element edit statement 10 is used to implement the element matching with one or more elements involved in a document portion 14 that is an document area to be searched. If the document portion 14 contains an element named <note>, a match is established.

On page 4, paragraph beginning on line 5, please amend as follows:

B4 The character pattern, which consists of normal text characters (such as typically used in UNIX commands) of a character sequence, is delimited or defined using the " (left double quotation mark) and " (right double quotation mark). The character pattern is used to search for document data which includes at least one character sequence defined by the character pattern. That is, the existence of document data is ascertained if a match is established between the character pattern and the document data. In the instant disclosure, the document data signifies a document portion excluding tags. One example of matching using character pattern is described with reference to Fig. 2. As shown in Fig. 2, the element edit statement 10 includes a character pattern "the". The structured document 12 involves an element defined by <ooo> and </ooo> which contains a character sequence of "the" and thus, a match is established between character pattern "the" and the document data within the element defined by <ooo> and </ooo>.

On page 5, paragraph beginning on line 4, please amend as follows:

B5 The negation indicator is defined by the '!' character (exclamation character). The negation indicator is used to specify an element wherein a match is not established with a character sequence immediately following the negation indicator. Fig. 4 is a diagram schematically showing one example of matching of a character sequence "this" preceded by the negation indictor with an element in a document portion 18 of the structured document 12. The character sequence ~~"the"~~ "this" is defined in the element edit statement 10.

On page 5, paragraph beginning on line 21, please amend as follows:

B6 The sequence connector is defined by the ',' (comma) character. ~~Condiering~~ Considering the example of <A>,<B>, in the case of which the element specified by tag A must precede the element specified by tag B. However, a match is also established even if

another element exists between the elements respectively defined tags A and B. One example of the sequence connector ',' is shown in Fig. 6. As shown, the element edit statement 10 contains the sequence connector between tags <beginning> and <ending>. The structured document 12 involves a document portion 20 which contains the elements "beginning" and "ending" in this order. As a result, matches are established as illustrated.

On page 6, paragraph beginning on line 18, please amend as follows:

The AND connector is defined using the '&' character. The example ~~'(E & F)~~ (E & F) indicates that 'F' may either follow or precede 'E' in the document portion to be searched for editing. Referring to Fig. 8, one example of usage of the AND connector is shown. As shown, the element edit statement 10 contains the AND connector '&' sandwiched by two tags <beginning> and <ending>. The structured document 12 contains the elements, defined by the tags <beginning> and <ending>, in the document portion 20 and thus, a successful match is established as illustrated.

On page 8, paragraph beginning on line 5, please amend as follows:

On the other hand, the input document 52 contains the following items.

Summary of paper

Name of Society: GHI Meeting

Title: Analysis of JKL

Name: Jiro ~~NAKAMURAO~~NAKAMURA

Abstract: This report, . . .

On page 8, paragraphs beginning on line 19, please amend as follows:

Fig. 12 shows the structured input documents 50' and 52' that respectively correspond to the input documents 50 and 52, while Fig. 13 shows the structured document 54' that ~~corresponding~~ corresponds to the output document 54. It is understood that the content of each of the items ("SUMMARY OF PAPER", "NAME OF SOCIETY", etc.) is defined using a start-tag defined by '<' and '>' and an end-tag defined by '</' and '>'. The detailed

B9 descriptions of Figs. 12 and 13 are deemed redundant and accordingly, will be omitted for brevity.

On page 8, paragraphs beginning on line 28, and ending on page 9, line 29, please amend as follows:

B10 Fig. 14 is a flow chart which shows the steps which characterize the overall operation of the document data editing according to the first embodiment. In Fig. 14, at step 60, the first input document 50 is applied to the memory 34 ~~form~~ from the database 36. Assuming that an element edit statement is:

%<title>,%<name> . . . element edit statement A

which is referred to as the element edit statement A for the sake of simplifying the discussion. At step 62, the element edit statement A is written into the document edit engine 42, after which the routine proceeds to a sub-routine 64 for element extraction. The sub-routine 64 is shown in Fig. 15 in detail.

Referring to Fig. 15, at step 66, if the element edit statement, retrieved into the document edit engine 42, contains one or more than one "OR connectors", the element edit statement is divided into a plurality of element edit instructions (operators), such as %<title> and %<name>, in every "OR connector". However, in the instant case, there exists no "OR connector" in the element edit statement A and thus, the step 66 is not executed. At step 68, one of the element edit instructions divided at step 66 is selected. However, as mentioned above, no division of the statement is carried out at step 66 and thus, the element edit statement A is selected as a whole, after which the routine goes to a sub-routine 70, the details of which is shown in Fig. 16.

Referring to Fig. 16, at step 72, the element edit statement containing "AND connector (&)" is processed. Although no "AND connector" is contained in the element edit

statement A, the flow chart of Fig 16 is described in ~~that~~ another example, set forth later, which includes the "AND connector". At step 72, if the element edit statement, applied to the document edit engine 42 (Fig. 10), contains one or more "AND connectors", the statement is divided into a plurality of element edit instructions or tags every each "AND connector". However, in the instant case, no "AND connector" is contained in the element edit statement A and thus, the step 72 is not executed. At step 74, one of the element edit instructions or tags, divided at step 72 is selected. However, as mentioned above, no division of the statement is carried out at step 72 and thus, the element edit statement A is selected as a whole, after which the program goes to a sub-routine 76, the details of which is shown in Fig. 17.

On page 9, paragraph beginning on line 30 and ending on page 11, line 5, please amend as follows:

In Fig. 17, at step 78, the first element edit instruction (or tag) of the element edit statement A is selected for execution. Following this, the program proceeds to step 80 at which a check is made to determine if the edit instruction (or tag) selected at step 78 (viz., <title> in the instant case) is parenthesized. The element edit tag <title> is not parenthesized and thus, the routine goes to step 82 at which the tag <title> is processed to determine if the element "title" matches any element of the structured document 50'. In the instant case, a match is established with the following element in the structured document 50':

<title>

DEF Report

</title>

Subsequently, at step 84, a check is made to determine if the matching execution has completed. In this case, since the answer to the inquiry made at step 84 is negative, the routine proceeds to step 86. The element edit statement contains the sequence connector ',' and hence, at step 86, the document portion following the element which has matched at step 82, becomes the next document portion to be searched. Following this, at step 88, the next element edit instruction (tag) (viz., <name>) is selected and the routine goes back to step 80. Thus, the following elements in the structured document 50' are matched:

<name>

Taro SATO </name>

B11

and

<name>

Hanako SUZUKI

</name>

Since the matching execution in connection with the document 50' completed, the answer to the inquiry made at step 84 is positive. Thus, the routine goes to step 90 at which a check is made to determine if any match has been established. In this case, the matches are established as mentioned above, the routine proceeds to step 92 at which a check is further made to determine if the extraction indicator '%' exists in the statement A. At step 94, since the extraction indicator '%' is involved in the element edit statement A, the matched elements are extracted and stored in the memory 44. Thereafter, the routine goes to step 96 of Fig. 16. In the above, if the answer at step 80 is positive, the parentheses are deleted at step 81 and the routine jumps to the sub-routine 64 (Fig. ~~15~~14).

On page 11, paragraph beginning on line 13, please amend as follows:

B12

Subsequently, the routine goes to 108 (Fig. 14) at which a check is made to determine if the element edit statement to be processed remains. Since no element edit statement to be processed with respect to the document 50' remains, the routine goes to step

110. The document ~~110~~ 52' has not yet been processed and hence, the routine goes back to step 60 at which the next document 52' is inputted to memory 34, after which the above mentioned processing is carried out in the same manner. The processing of the document 52' is clear from the foregoing, the description thereof will be omitted for brevity. It is understood that the element extracted from the document 52' is:

B12

<title>

ANALYSIS OF JKL

</title>

and

<name>

On page 12, paragraph beginning on line 27, please amend as follows:

B13

A third example of the first embodiment will be described, which includes OR connector. An input document, indicated as a structured document, is shown in Fig. 18 and denoted by numeral 120. As shown in Fig. 18, an outermost element is a paper element which contains a first-p (first paragraph) element and a second-p (second paragraph). Further, the first-p element contains a figure element, and similarly, the second-p element contains a figure element. The third example is to extract the figure element from each of the first-p and second-p. Thus, the element edit statement in the third example is:

(<first-p> | <second-p>)%<figure> . . . element edit statement C

As shown, this statement is referred to as "element edit statement C" for the sake of convenience of description. It is to be noted that (<first-p> | <second-p>) and %<figure> are connected by the hierarchy connector since no character is provided therebetween. The third

example contains OR connector and thus, the tags within the parentheses are divided at step 66 (Fig. 15), after which these tags are separated and are subject to the element matching process on an element-by-element basis. As mentioned above, since the hierarchy connection is used, the element <figure> of each of the elements <first-p> and <second-p> is extracted.

B13

On page 14, paragraph beginning on line 10, please amend as follows:

Fig. 21 shows a structured ~~out~~ output document 150' which corresponds to the output document 150 of Fig. 20.

B14

On page 14, paragraph beginning on line 12, please amend as follows:

B15

In order to generate the list C shown in Fig. 20, element edit statements D-1 and D-2 are inputted to the memory 38. More specifically,

The element edit statement D-1 contains:

(1) Edit statement 1 which is %<title> for extracting the element <title>;

(2) Edit statement 2 for changing the name "title" to "~~Titel~~ Title of Paper" and storing the changed name in a variable "title".

The element edit statement D-2 contains:

(1) Edit statement 3 which is %<name> for counting the number of authors;

(2) Edit statement 4 which counts the number of extracted names and stores the counted number in a variable "count".